



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Nuxoll et al.**

Serial No.: 09/215,752

Filed: December 18, 1998

For: **Method and Apparatus for a  
Meta Data Service in a Data  
Processing System**

§  
§  
§  
§  
§

Group Art Unit: 2151

Examiner: **Courtenay III, S.**

Attorney Docket No.: AT9-98-464

RECEIVED

MAY 17 2001

Technology Center 2100

Certificate of Mailing Under 37 C.F.R. § 1.8(a)

I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Assistant Commissioner of Patents, Washington, D.C. 20231 on May 14, 2001.

By:

*Amelia C. Nearing*  
Amelia C. Nearing

TRANSMITTAL DOCUMENT

Assistant Commissioner of Patents  
Washington, D.C. 20231

Sir:

ENCLOSED HEREWITH:

- Appellant's Brief (in triplicate) (37 C.F.R. 1.192); and
- Our return postcard.

A fee of \$310.00 is required for filing an Appellant's Brief. **Please charge this fee to IBM Deposit Account No. 09-0447.** No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to **IBM Deposit Account No. 09-0447.** No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to Deposit Account No. 50-0392.

Respectfully submitted,

*Duke W. Yee*  
Duke W. Yee

Registration No. 34,285

CARSTENS, YEE & CAHOON, LLP

P.O. Box 802334

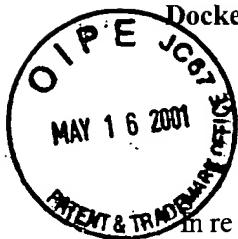
Dallas, Texas 75380

(972) 367-2001

ATTORNEY FOR APPLICANTS

Docket No. AT9-98-464

PATENT



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

RECEIVED

MAY 17 2001

Technology Center 2100

In re application of: **Nuxoll et al.**

Serial No.: **09/215,752**

Filed: **December 18, 1998**

For: **Method and Apparatus for a Meta  
Data Service in a Data Processing System**

§  
§  
§  
§  
§  
§  
§

Group Art Unit: **2151**

Examiner: **Courtenay III, S.**

Assistant Commissioner for Patents  
Washington, D.C. 20231

ATTENTION: Board of Patent Appeals and  
Interferences

Certificate of Mailing Under 37 C.F.R. § 1.8(a)

I hereby certify this correspondence is being deposited with the  
United States Postal Service as First Class mail in an envelope  
addressed to: Assistant Commissioner of Patents, Washington,  
D.C. 20231 on May 14th, 2001.

By:

*Amelia C. Nearing*  
Amelia C. Nearing

**APPELLANT'S BRIEF (37 C.F.R. 1.192)**

This brief is in furtherance of the Notice of Appeal, filed in this case on 14 March 2001.

The fees required under § 1.17(c), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate. (37 C.F.R. 1.192(a))

05/17/2001 GTEFFERA 00000097 090447 09215752

01 FC:120 310.00 CH

**REAL PARTIES IN INTEREST**

The real party in interest in this appeal is the following party: IBM Corporation

RECEIVED  
MAY 17 2001  
Technology Center 2100

**RELATED APPEALS AND INTERFERENCES**

With respect to other appeals or interference's that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interference's.

**STATUS OF CLAIMS**

**A. TOTAL NUMBER OF CLAIMS IN APPLICATION**

Claims in the application are: 1, 2, 4-7, 10-14, 16, 18-20, 22, and 24-47

**B. STATUS OF ALL THE CLAIMS IN APPLICATION**

1. Claims canceled: 3, 8, 9, 15, 17, 21, and 23
2. Claims withdrawn from consideration but not canceled: NONE
3. Claims pending: 1, 2, 4-7, 10-14, 16, 18-20, 22, and 24-27
4. Claims allowed: NONE
5. Claims rejected: 1, 2, 4-7, 10-14, 16, 18-20, 22, and 24-27

**C. CLAIMS ON APPEAL**

The claims on appeal are: 1, 2, 4-7, 10-14, 16, 18-20, 22, and 24-27

RECEIVED

MAY 17 2001

Technology Center 2100

### STATUS OF AMENDMENTS

In response filed 28 February 2001, Appellant proposed amendment to claim 34 to overcome the objection to the claim due to incorrect dependency. Appellant received acknowledgement that the response was received by the Office; however, an Office Action has not been received. Appellant assumes that the amendment will be entered, because it simplifies and reduces the issues for appeal.

### SUMMARY OF INVENTION

The present invention provides a Meta Data Service in a network that stores definitions of all objects that are used by applications that are designed to use the Meta Data Service. A client queries the Meta Data Service for object definitions for objects and creates a GUI for a user to supply attribute information. See specification, page 11, lines 10-16. After the user enters the data, the client submits the object for processing by one or more servers in the network by transmitting the object as a stream of attribute data values. See specification, page 12, lines 1-4. Thus, the client does not need to be specifically programmed for each and every possible object. The client queries the Meta Data Service for object definitions to create the GUI and transmits the data in a "soft format" to the servers. The servers receive the stream of attribute data values and query the Meta Data Service for a definition of the object. See specification, page 12, lines 5-12. The server may then perform any necessary attribute processing and transmit the object as a data value stream to a Persistent Object Service. See specification, page 12, lines 17-26. A server does not need to be specifically programmed for each and every possible object, because the server is programmed to query the Meta Data Service for object definitions. The Persistent Object Service receives the object in a soft format, the data value stream. The Persistent Object Service then queries the Meta Data Service for an object definition. The Persistent Object Service may then map data values to relational tables and store the object in a relational database management system. See specification, page 12, line 27, to page 13, line 16. Again, like the client and server, the Persistent Object Service does not need to be specifically programmed for each object. Objects may be transmitted through the network in a soft format and each node may query the Meta Data Service for an object definition.

## ISSUES

The issues on appeal are as follows:

Whether claims 1, 2, 5, 7, 10, 13, 14, 16, 18-20, 22, 24-35, 37, 40, 42, and 46 are unpatentable as being anticipated by Maruyama et al.; and

Whether claims 4, 6, 11, 12, 36, 38, 39, 41, 43, 44, 45, and 47 are unpatentable as being obvious over Maruyama et al. in view of allegedly well known prior art.

## GROUPING OF CLAIMS

The claims on appeal do not stand or fall in a single group, but are grouped into the following groups:

Claims 1, 13, and 19 form group A. Claims 2, 14, 20, and 37 form group B. Claims 4, 11, 38, 39, 44, and 45 form group C. Claims 5 and 40 form group D. Claims 6, 12, and 41 form group E. Claims 7, 16, 18, 22, 24, 27, 28, 32, and 33 form group F. Claim 10 forms group G. Claims 25 and 30 form group H. Claims 26 and 31 form group I. Claims 29 and 34 form Group J. Claims 35, 42, and 46 form group K. And, claims 36, 43, and 47 form group L.

## ARGUMENT

### **I. Claim Objection**

The Office Action objects to claim 34 because it depends on itself. Claim 34, as amended, now depends on claim 33. Therefore, the objection is overcome.

### **II. 35 U.S.C. § 102, Anticipation**

The Office Action rejects claims 1, 2, 5, 7, 10, 13, 14, 16, 18-20, 22, 24-35, 37, 40, 42, and 46 under 35 U.S.C. § 102 as being anticipated by Maruyama et al. (US Patent No. 5,710,920), hereinafter referred to as "*Maruyama*." This rejection is respectfully traversed.

*Maruyama* discloses a method for extending objects in an object-oriented database. The system and method of *Maruyama* permit an object to be changed in terms of attribute, relation, and procedure independently of schema definition information in the object-oriented database. In contradistinction, the

present invention concerns separating the meta data from application code within a distributed data processing system.

In accordance with the present invention, a client is used to input attributes for an object without the software in the client being dependent on the meta data for the object. The software in the client that assists the user in creating a data object is not programmed with the meta data that defines the structure of the object. Instead, the client software must receive the meta data or meta definition from a Meta Data Service in order to, for example, generate graphical user interface fields. This aspect of the invention is covered by claims 1, 2, 4-6, 13, 14, 19, 20, 25, 26, 30, and 31.

Also, in accordance with the present invention, a server in a data processing system may receive data objects in a “soft” format. This soft format comprises a data value stream without meta data that defines the structure of the object. Applications in the data processing system are designed in such a manner that they are not presumed to understand the definition of the data object. Thus, a server must query a Meta Data Service for a meta definition. This aspect of the invention is covered by claims 35-47.

Further, in accordance with the present invention, a Persistent Object Service also receives objects in a soft format. The software in the Persistent Object Service also is not presumed to understand the definition of the data object. Therefore, the Persistent Object Service must query the Meta Data Service for a meta definition and map attributes in the received data streams with attributes in the meta definition of the data object before storing the object to persistent storage. This aspect of the invention is covered by claims 7, 10-12, 16, 18, 22, 24, 27-29, and 32-34.

With respect to claims 1, 13, 19, and 42, the Office Action states:

**As per independent claims 1, 13, 19, 42:**

**Maruyama** teaches a method in a software component for processing a data object in a data processing system, the method comprising the computer-implemented steps of:

- sending a query for a meta definition of a data object [e.g., step 1001, fig. 13, col. 9, lines 37-45],
- receiving the meta definition of the data object [e.g., col. 9, line 45].
- identifying object attributes in the meta definition [e.g., col. 9, lines 45-64], line ].
- prompting a user to input data values corresponding to the object attributes [e.g., col. 9, line 38].

Office Action dated 14 December 2000. Appellant respectfully disagrees. *Maruyama* teaches a dictionary **108** storing type definition information **within an object-oriented database**. However, *Maruyama* does not teach “identifying attributes in the meta definition” and “prompting a user to input data values corresponding to **the object attributes**” [emphasis added], as recited in claims 1, 13, and 19.

The cited text of *Maruyama* is reproduced as follows:

**FIG. 13** is a flow for processing of reflecting a change in type definition when restructuring the DB. The user designates an object identifier of a type definition object to the object manger 103 through the view manager 101. In **FIG. 13**, steps 1001, 1004 and 1008 are processed by the dictionary manager 104 and the other steps are processed by the object manager 103.

In step 1001, the type definition object is acquired. The following processing is carried out until information of parts attribute object is determined to be absent in step 1002. In step 1003, it is examined whether the processing for all change information pieces owned by the parts attribute object ends. If unfinished, attribute change information is acquired from the parts object manager 102 in step 1004 and on the basis of this information, attribute information owned by the type definition object of schema is changed. In step 1005, all objects belonging to that type are changed on the basis of the type information after change. At that time, attribute data owned by a structure change object is used. In step 1006, it is examined whether a parts procedure object having procedure change information is present. If present, it is examined in step 1007 whether the processing for all change information pieces owned by the parts procedure object ends. If absent, a parts procedure object is acquired from the parts object manager 102 and on the basis of this information, procedure information of the type definition object in the schema is changed in step 1008.

*Maruyama*, col. 9, lines 37-64. According to the cited portion of the reference, “[t]he user designates an object identifier of a type definition object.” However, *Maruyama* does not teach or suggest identifying object attributes in the meta definition and prompting the user to input data values corresponding to the **identified** object attributes, as recited in claims, 1, 13, and 19. Each and every claim element is not taught by the prior art; therefore, the claimed invention is not anticipated by the reference.

Present claim 42 recites:

42. A data processing system for processing a data object, said data processing system comprising:

- first receipt means for receiving a data value stream for a data object;
- sender means for sending a query for a meta definition of a data object;
- second receipt means for receiving the meta definition of the data object; and
- process means for processing the data object according to attributes in the meta definition for the data object to form a second data value stream.

*Maruyama* does not teach or suggest “first receipt means for receiving a data value stream for a data object” or “process means for processing the data object according to attributes in the meta definition for the data object to form a second data value stream,” as recited in claim 42. As discussed above, *Maruyama* is concerned only with an object-oriented database and not the manner in which objects are processed in a distributed data processing system. Therefore, *Maruyama* does not teach or suggest the data processing

system recited in claim 42 in which an application receives an object in a soft format (data value stream) and queries a Meta Data Service for the meta definition. The Office Action does not address these features; therefore, the rejection must be withdrawn.

With respect to claims 7, 16, 22, 35, and 46, the Office Action states:

**As per independent claims 7, 16, 22, 35, 46:**

**Maruyama** teaches a method in a software component for processing a data object in a data processing system, the method comprising the computer-implemented steps of:

- receiving a data value stream [e.g., col. 9, also figs. 12 & 13].
- sending a query for a meta definition of a data object [e.g., step 1001, fig. 13, col. 9, lines 37-45],
- receiving the meta definition of the data object [e.g., col. 9, line 45].
- mapping data values to a data structure according to the attributes in the meta definition of the data object [e.g., col. 9, lines 45-64].

Office Action dated 14 December 2000. Appellant respectfully disagrees. Again, *Maruyama* is concerned only with an object-oriented database and not the manner in which objects are processed in a distributed data processing system. Therefore, *Maruyama* does not teach or suggest the data processing system recited in claims 7, 16, and 22, in which an application receives an object in a soft format (data value stream) and queries a Meta Data Service for the meta definition. The Office Action refers to an entire column of the reference and two flowcharts; however, it is unclear where the step of receiving a data value stream is taught. Furthermore, the cited portion of the reference does not disclose mapping data values to a data structure according to attributes in a received meta definition. *Maruyama* only teaches managing changes to data type definitions within an object-oriented database. *Maruyama* does not teach that the database receives objects in a “soft” format, i.e. in a data value stream. Each and every claim element is not taught by the prior art; therefore, the claimed invention is not anticipated by the reference.

Claims 35 and 46 recite:

35. A method in a software component for processing a data object in a data processing system, said method comprising the computer-implemented steps of:

- receiving a first data value stream for a data object;
- sending a query for a meta definition of the data object;
- receiving a meta definition of the data object; and
- processing the data object according to attributes in the meta definition of the data object to form a second data value stream for the data object.

46. A computer program product for use with a data processing system for processing a data object, said computer program product comprising:

- a computer readable medium;
- first instructions for receiving a first data value stream for a data object;
- second instructions for sending a query for a meta definition of the data object;



third instructions for receiving a meta definition of the data object; and  
fourth instructions for processing the data object according to attributes in the  
meta definition of the data object to form a second data value stream for the data object.

*Maruyama* does not teach or suggest “receiving a data value stream for a data object” or “processing the data object according to attributes in the meta definition for the data object to form a second data value stream,” as recited in claims 35 and 46. As discussed above, *Maruyama* is concerned only with an object-oriented database and not the manner in which objects are processed in a distributed data processing system. Therefore, *Maruyama* does not teach or suggest the method and computer program product recited in claims 35 and 46 in which a software component receives an object in a soft format (data value stream) and queries a Meta Data Service for the meta definition. The Office Action does not address these features; therefore, the rejection must be withdrawn.

Since claims 2, 5, 10, 14, 18, 20, 24-34, 37, and 40 depend from claims 1, 7, 13, 16, 19, 22, and 35, the same distinctions between *Maruyama* and the claimed invention in claim 1, 7, 13, 16, 19, 22, and 35 for these claims. Additionally, claims 2, 5, 10, 14, 18, 20, 24-34, 37, and 40 claim other additional combinations of features not suggested by the reference.

Particularly, with respect to claims 2, 14, 20, and 37, the Office Action states:

**As per claims 2, 14, 20, 37:**

**Maruyama** teaches determining an object type of a data object, before querying for a meta definition for the data object [e.g., col. 3, see discussion beginning line 11, meta definition discussion begins line 61].

Office Action dated 14 December 2000. Appellant respectfully disagrees. *Maruyama* teaches that information defined by a type is a meta definition. However, *Maruyama* does not teach a method, apparatus, or program for querying for a meta definition of a data object to process a data value stream. *Maruyama* does not teach a method, apparatus, or program for receiving or transmitting data values in a soft format; therefore, there is no need to determine the object type in *Maruyama*.

With respect to claims 5 and 40, the Office Action states:

**As per claims 5, 40:**

**Maruyama** teaches the query for the meta definition for the data object is sent to a Meta Data Service [e.g., step 1001, fig. 13, col. 9, lines 37-45, col. 3].

Office Action dated 14 December 2000. Appellant respectfully disagrees. *Maruyama* teaches an object-oriented database. Specifically, *Maruyama* teaches an object-oriented database in which an object may be changed in terms of attribute, relation, and procedure independently of schema. See *Maruyama*, col. 1, lines 58-62. Meta data is stored in a database for objects in the object-oriented database. *Maruyama* does not teach or suggest applications in a network for receiving or transmitting attribute value data in a

soft format. Therefore, there is no need in *Maruyama* for a Meta Data Service as claimed and described in the present specification.

With respect to claim 10, the Office Action states:

**As per claim 10:**

**Maruyama** teaches the software component is a Meta Data Service. [e.g., see discussion columns 3 & 4].

Office Action dated 14 December 2000. Claim 10 recites:

10. The method according to claim 7 wherein the software component is a Persistent Object Service.

The rejection of claim 10 is unclear. *Maruyama* does not teach the steps recited in claim 7; therefore, *Maruyama* cannot teach that the software component is a Persistent Object Service, as recited in claim 10 and described by the present specification. *Maruyama* also fails to teach or suggest a Meta Data Service. However, whether or not *Maruyama* teaches a Meta Data Service is irrelevant because claim 10 does not recite a Meta Data Service.

With respect to claims 26 and 31, the Office Action states:

**As per claim 26:**

**Maruyama** teaches the step of prompting the user for data values comprises: matching the meta definition to graphical user interface fields; and presenting the graphical user interface fields to the user [e.g., col. 7, lines 60-65, col. 9, line 5, line 39].

**As per claim 31:**

**Maruyama** teaches the prompting means comprises: means for matching the meta definition to graphical user interface fields; and means for presenting the graphical user interface fields to the user [e.g., col. 7, lines 60-65, col. 9, line 5, line 39].

Office Action dated 14 December 2000. Appellant respectfully disagrees. The cited portion of *Maruyama* states:

**FIG. 10** is a flow for processing of attribute change. When requesting the present processing, the user transfers an object identifier, a version number and a change request code (appending, deletion and update) upon definition change and change attribute information including attribute name, attribute type, type size and attribute value to the object manager 103 through the view manager 101.

*Maruyama*, col. 7, lines 60-66.

When requesting the present processing, the user designates an object identifier, a change request code (appending, deletion and update), a selector name, a parameter group (type, value) and an execution code to the object manager 103 through the view manager 101.

*Maruyama*, col. 9, lines 5-9.

The user designates an object identifier of a type definition object to the object manger **103** through the view manager **101**.

*Maruyama*, col. 9, lines 38-40. The cited passages teach that a user designates an object identifier. However, the reference does not teach or suggest “matching the meta definition to graphical user interface fields” or “presenting the graphical user interface fields to the user,” as recited in claim 26. Each and every claim limitation is not taught by the reference; therefore, *Maruyama* does not anticipate the claim. Claim 31 is allowable for the same reasons.

With respect to claims 25 and 30, the Office Action states:

**As per claims 25, 30:**

**Maruyama** teaches means for receiving inputted data values corresponding to the object attributes from the user; and means for sending a data value stream including the inputted data values to a server [e.g., col. 7, lines 60-65, col. 9, line 5, line 39].

Office Action dated 14 December 2000. Appellant respectfully disagrees. The cited passages teach that a user designates an object identifier. However, the reference does not teach or suggest “receiving inputted data values corresponding to the object attributes from the user” and “sending a data value stream including the inputted data values to a server,” as recited in claims 25 and 30. As discussed above, *Maruyama* is concerned only with an object-oriented database and not the manner in which objects are processed in a distributed data processing system. Therefore, *Maruyama* does not teach or suggest the method and data processing system recited in claims 25 and 30 in which a software component queries a Meta Data Service for the meta definition, receives attributes in a soft format (data value stream), and sends the attributes to a server in a soft format.

With respect to claims 29 and 34, the Office Action states:

**As per claims 29, 34:**

**Maruyama** teaches the database is a relational database [e.g., col. 3, lines 5-7].

Office Action dated 14 December 2000. Appellant respectfully disagrees. The cited portion of *Maruyama* states:

A relation between objects is held in the form of a relation definition. The relation definition is determined definitely in a schema and is identified by a relation name. The attribute, relation and procedure are collectively called a "characteristic" of the object.

*Maruyama*, col. 3, lines 5-10. The reference states that a relation between objects is held as a

characteristic of the object; however, the reference does not teach a relational database. To the contrary, *Maruyama* specifically relates to an object-oriented database system. See *Maruyama*, col. 1, lines 1-7. Each and every claim limitation is not taught by the prior art. Therefore, *Maruyama* does not anticipate claims 29 and 34.

Therefore, the rejection of claims 1, 2, 5, 7, 10, 13, 14, 16, 18-20, 22, 24-35, 37, 40, 42, and 46 under 35 U.S.C. § 102 is overcome.

Furthermore, *Maruyama* does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. Absent some teaching or incentive to implement *Maruyama* in a distributed data processing system in which software components pass data objects as data value streams and query a Meta Data Service for meta definitions, one of ordinary skill in the art would not be led to modify *Maruyama* to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify *Maruyama* in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the appellants' disclosure as a template to make the necessary changes to reach the claimed invention.

### **III. 35 U.S.C. § 103, Obviousness**

The Office Action rejects claims 4, 6, 11, 12, 36, 38, 39, 41, 43-45, and 47 under 35 U.S.C. § 103 as being unpatentable over *Maruyama* in view of well known prior art. This rejection is respectfully traversed.

Since claims 4, 6, 11, 12, 36, 38, 39, 41, 43-45, and 47 depend from claims 1, 7, 35, 42, and 46, the same distinctions between *Maruyama* and the claimed invention in claims 1, 7, 13, 16, 19, 22, and 35 apply for these claims. Additionally, claims 2, 5, 10, 14, 18, 20, 24-34, 37, and 40 claim other additional combinations of features not suggested by the reference. Therefore, the rejection of claims 4, 6, 11, 12, 36, 38, 39, 41, 43-45, and 47 under 35 U.S.C. § 103 is overcome.

Particularly, with respect to claims 6 and 41, the Office Action states:

**As per claims 6, 41:**

**Maruyama** discloses the invention substantially as claimed, as discussed above.

However, **Maruyama** does not *explicitly* teach the use of a cache.

“Official Notice” is taken that the use of caches is well known in the art [M.P.E.P. 2144.03].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to improve upon the system taught by **Maruyama** by implementing the improvements detailed above because it would provide **Maruyama's** system with the

enhanced capability of improving performance.

Office Action dated 14 December 2000. Appellant respectfully disagrees. *Maruyama* does not teach or suggest a method, apparatus, or program wherein attribute value data is received or transmitted in a soft format. Therefore, there is no need in *Maruyama* for a Meta Data Service. Hence, *Maruyama* provides no suggestion that querying a Meta Data Service for a meta definition and receiving a meta definition from a Meta Data Service requires a cache or that performance would be improved by use of a cache. The Office Action may not make modifications to the prior art using the claimed invention as a model for the modifications. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780, 1783-1784 (Fed. Cir. 1992). "The mere fact that the prior art may be modified in the manner suggested by the Examiner does not make the modification obvious unless the prior art has suggested the desirability of the modification." *Id.* In other words, unless some teaching exists in the prior art for the suggested modification, merely asserting that such a modification would be obvious to one of ordinary skill in the art is improper and cannot be used to meet the burden of establishing a *prima facie* case of obviousness. Such reliance is an impermissible use of hindsight with the benefit of applicant's disclosure.

Therefore, absent some teaching, suggestion, or incentive in the prior art, *Maruyama* cannot be properly modified to form the claimed invention. As a result, absent any teaching, suggestion, or incentive from the prior art to make the proposed modifications, the presently claimed invention can be reached only through an impermissible use of hindsight with the benefit of applicant's invention as a model. Claim 12 is allowable for the same reasons as claims 6 and 41.

With respect to claims 36, 43, and 47, the Office Action states:

**As per claims 36, 43, 47:**

**Maruyama** discloses the invention substantially as claimed, as discussed above.

However, **Maruyama** does not *explicitly* teach the use of Persistent Objects and associated services.

"Official Notice" is taken that the use of persistent objects and associated services is well known in the art [M.P.E.P. 2144.03].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to improve upon the system taught by **Maruyama** by implementing the improvements detailed above because it would provide **Maruyama's** system with the enhanced capability of persistent data storage.

Office Action dated 14 December 2000. Appellant respectfully disagrees. *Maruyama* teaches an object-oriented database. *Maruyama* does not teach a method, apparatus, or program for processing a data object and transmitting a data value stream to a Persistent Object Service. The object-oriented database

of *Maruyama* would not be able to store data values received in a soft format. Hence, it would not have been obvious to transmit the second data value stream to a Persistent Object Service. Therefore, absent some teaching, suggestion, or incentive in the prior art, *Maruyama* cannot be properly modified to form the claimed invention. As a result, absent any teaching, suggestion, or incentive from the prior art to make the proposed modifications, the presently claimed invention can be reached only through an impermissible use of hindsight with the benefit of applicant's invention as a model.

With respect to claims 4, 11, 38, 39, 44, and 45, the Office Action states:

**As per claims 4, 11, 12, 38, 39, 44, 45:**

**Maruyama** discloses the invention substantially as claimed, as discussed above.

However, **Maruyama** does not *explicitly* teach the use of clients and servers. *Maruyama* does teach the use of multiple computers in an object database system [e.g., col. 6, lines 23, 61]

“Official Notice” is taken that the use of clients and servers is notoriously well known in the art [M.P.E.P. 2144.03].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to improve upon the system taught by **Maruyama** by implementing the improvements detailed above because it would provide a distributed object database system.

Office Action dated 14 December 2000. Appellant respectfully disagrees. The cited portions of *Maruyama* state the phrases “computer type” and “computer object”; however, *Maruyama* provides no suggestion for implementing a distributed object database system. More particularly, *Maruyama* provides not suggestion for implementing a software component in a client for prompting a user to enter object attribute values, as specifically recited, in combination, in claim 4. Furthermore, *Maruyama* provides no suggestion for sending a query for a meta definition to a Meta Data Service, as specifically recited, in combination, in claim 11. Still further, *Maruyama* provides no suggestion for implementing a software component in a first server for receiving a data value stream, sending a query for a meta definition, receiving a meta definition, and processing the data value stream, as specifically recited, in combination, in claims 38 and 44. Hence, *Maruyama* cannot teach the further step of transmitting a second data value stream to a second server, as specifically recited, in combination, in claims 39 and 45.

Therefore, absent some teaching, suggestion, or incentive in the prior art, *Maruyama* cannot be properly modified to form the claimed invention. As a result, absent any teaching, suggestion, or incentive from the prior art to make the proposed modifications, the presently claimed invention can be reached only through an impermissible use of hindsight with the benefit of applicant's invention as a model.

IV. Conclusion

In view of the above comments, it is respectfully urged that the rejection of the claims not be sustained.

DATE: 14 May 2001

Respectfully submitted,



Stephen R. Tkacs  
Reg. No. 46,430  
Carstens, Yee & Cahoon, LLP  
PO Box 802334  
Dallas, TX 75380  
(972) 367-2001

## APPENDIX OF CLAIMS

The text of the claims involved in the appeal are:

1. A method in a software component for processing a data object in a data processing system, said method comprising the computer-implemented steps of:
  - sending a query for a meta definition of a data object;
  - receiving the meta definition for the data object;
  - identifying object attributes in the meta definition; and
  - prompting a user to input data values corresponding to the object attributes.
2. The method according to claim 1 further comprising, before querying for a meta definition of a data object, determining an object type of a data object.
3. The method according to claim 1 further comprising receiving the meta definition for the data object prior to the step of processing the data object.
4. The method according to claim 1 wherein the software component is in a client.
5. The method according to claim 1 wherein the query for the meta definition for the data object is sent to a Meta Data Service.
6. The method according to claim 5 wherein the meta definition is cached to improve the performance of the Meta Data Service.
7. A method in a software component for processing a data object in a data processing system, said method comprising the computer-implemented steps of:
  - receiving a data value stream;
  - sending a query for a meta definition of a data object;
  - receiving a meta definition of the data object; and
  - mapping data values to a data structure according to attributes in the meta definition of the data object.



8. A method according to claim 7 further comprising storing at least one meta definition of a data object.
9. A method according to claim 7 further comprising at least one object attribute in the meta definition of the data object.
10. The method according to claim 7 wherein the software component is a Persistent Object Service.
11. The method according to claim 7 wherein the query for the meta definition for the data object is sent to a Meta Data Service.
12. The method according to claim 11 wherein the meta definition is cached to improve the performance of the Meta Data Service.
13. A data processing system for processing a data object, said data processing system comprising:  
querying means for querying for a meta definition of a data object;  
receipt means for receiving the meta definition for the data object;  
identification means for identifying object attributes in the meta definition; and  
prompting means for prompting a user to input data values corresponding to the object attributes.
14. The data processing system according to claim 13 further comprising determination means for determining an object type of a data object before querying a Meta Data Service for a meta definition of a data object.
15. The data processing system according to claim 13 further comprising receipt means for receiving the meta definition for the data object prior to processing the data object.
16. A data processing system for processing a data object, said data processing system comprising:  
first receipt means for receiving a data value stream;  
sender means for sending a query for a meta definition of a data object;  
second receipt means for receiving the meta definition of the data object ; and  
mapping means for mapping data values to a data structure according to attributes in the meta definition of the data object.

17. A data processing system according to claim 16 further comprising storage means for storing at least one meta definition of a data object.

18. A data processing system according to claim 16 further comprising at least one object attribute in the meta definition of the data object.

19. A computer program product for use with a data processing system for processing a data object, said computer program product comprising:

- a computer usable medium;
- first instructions for sending a query for a meta definition of a data object;
- second instructions for receiving the meta definition for the data object;
- third instructions for identifying object attributes in the meta definition; and
- fourth instructions for prompting a user to input data values corresponding to the object attributes.

20. The computer program product according to claim 19 further comprising instructions for determining an object type of a data object before querying a Meta Data Service for a meta definition of a data object.

21. The computer program product according to claim 19 further comprising instructions for receiving the meta definition for the data object prior to processing the data object.

22. A computer program product for use with a data processing system for processing a data object, said computer program product comprising:

- a computer usable medium;
- first instructions for receiving a data value stream;
- second instructions for sending a query for a meta definition of a data object;
- third instructions for receiving the meta definition of the data object; and
- fourth instructions for mapping data values to a data structure according to attributes in the meta definition of the data object.

23. A computer program product according to claim 22 further comprising instructions for storing at least one meta definition of a data object.

24. A computer program product according to claim 22 further comprising at least one object attribute in the meta definition of the data object.
25. The method of claim 1, further comprising:  
receiving inputted data values corresponding to the object attributes from the user; and  
sending a data value stream including the inputted data values to a server.
26. The method of claim 1, wherein the step of prompting the user for data values comprises:  
matching the meta definition to graphical user interface fields; and  
presenting the graphical user interface fields to the user.
27. The method of claim 7, further comprising transferring the data values to the data structure.
28. The method of claim 7, wherein the data structure is a database.
29. The method of claim 28, wherein the database is a relational database.
30. The system of claim 13, further comprising:  
means for receiving inputted data values corresponding to the object attributes from the user; and  
means for sending a data value stream including the inputted data values to a server.
31. The system of claim 13, wherein the prompting means comprises:  
means for matching the meta definition to graphical user interface fields; and  
means for presenting the graphical user interface fields to the user.
32. The system of claim 16, further comprising transfer means for transferring the data values to the data structure.
33. The system of claim 16, wherein the data structure is a database.
34. The method of claim 33, wherein the database is a relational database.

35. A method in a software component for processing a data object in a data processing system, said method comprising the computer-implemented steps of:
- receiving a first data value stream for a data object;
  - sending a query for a meta definition of the data object;
  - receiving a meta definition of the data object; and
  - processing the data object according to attributes in the meta definition of the data object to form a second data value stream for the data object.
36. The method of claim 35, further comprising transferring the second data value stream to a Persistent Object Service.
37. The method of claim 35, further comprising, before querying for a meta definition of a data object, determining an object type of a data object.
38. The method of claim 35, wherein the software component is in a first server.
39. The method of claim 38, further comprising transferring the second data value stream to a second server.
40. The method of claim 35, wherein the query for the meta definition for the data object is sent to a Meta Data Service.
41. The method of claim 40, wherein the meta definition is cached to improve the performance of the Meta Data Service.
42. A data processing system for processing a data object, said data processing system comprising:
- first receipt means for receiving a data value stream for a data object;
  - sender means for sending a query for a meta definition of a data object;
  - second receipt means for receiving the meta definition of the data object; and
  - process means for processing the data object according to attributes in the meta definition for the data object to form a second data value stream.

43. The system of claim 42, further comprising transfer means for transferring the second data value stream to a Persistent Object Service.

44. The method of claim 42, wherein the software component is in a first server.

45. The method of claim 44, further comprising transfer means for transferring the second data stream to a second server.

46. A computer program product for use with a data processing system for processing a data object, said computer program product comprising:

- a computer usable medium;
- first instructions for receiving a data value stream for a data object;
- second instructions for sending a query for a meta definition of a data object;
- third instructions for receiving the meta definition of the data object; and
- fourth instructions for processing the data object according to attributes in the meta definition for the data object to form a second data value stream.

47. The system of claim 46, further comprising fifth instruction means for transferring the second data value stream to a Persistent Object Service.